



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ŠACHOVÝ PROGRAM S RÔZNYMI VARIANTAMI ŠACHU S ROZDIELNOU SADOU FIGÚR

CHESS PROGRAM WITH VARIOUS CHESS VARIATIONS WITH VARIOUS SET OF FIGURES

BAKALÁRSKÁ PRÁCA

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FRANTIŠEK ŠKANDERA

VEDÚCI PRÁCE

SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2012

Abstrakt

Cieľom tejto práce bolo vytvoriť šachový program so schopnosťou hry rôznych šachových variácií s rozdielnou sadou figúr. Prvá časť práce sa zaoberá všeobecnou problematikou tvorby šachového programu a implementácie šachovej umelej inteligencie. V druhej časti analyzujeme jednotlivé zvolené šachové varianty a navrhujeme stratégie pre ich implementáciu. Nakoniec popisujeme samotnú implementáciu vytvoreného programu a jeho štruktúru.

Abstract

The goal of this thesis was to create a chess program able to play various chess variations with various set of figures. The first part of the work deals with the general matters of creating a chess program and implementing a chess artificial intelligence. In the second part we analyze individual chosen chess variations and suggest strategies for their implementation. Eventually we describe the implementation of the created application itself and its structure.

Klíčové slová

šach, program, rozdielna sada figúr, umelá inteligencia, prehľadávanie stavového priestoru, minimax, alfabeta, ohodnocovacia funkcia

Keywords

chess, program, various set of figures, artificial intelligence, state space search, minimax, alpha-beta, evaluation function

Citácia

František Škandera: Šachový program s rôznymi variantami
šachu s rozdielnou sadou figúr, bakalárska práca, Brno, FIT VUT v Brně, 2012

Šachový program s rôznymi variantami šachu s rozdielnou sadou figúr

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Jaroslava Rozmana, PhD.

.....
František Škandera
16. května 2012

PodĎakovanie

Týmto by som chcel poďakovať Ing. Jaroslavovi Rozmanovi, PhD. za trpezlivý prístup pri vypracúvaní tejto bakalárskej práce, a rodine a priateľom za morálnu podporu.

© František Škandera, 2012.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Šach	3
2.1	História šachu	3
2.2	Pravidlá šachu	3
3	Tvorba šachového programu	6
3.1	Spôsob reprezentácie hernej plochy	6
3.2	Prehľadávanie stavového priestoru	7
3.2.1	MiniMax	7
3.2.2	Rozšírenie Alfa-Beta	8
3.2.3	Heuristiky pre zefektívnenie MiniMaxu a Alfa-Bety	10
3.3	Ohodnocovacia funkcia	10
3.3.1	Materiálna zložka	11
3.3.2	Pozičná zložka	11
4	Zvolené šachové varianty	14
4.1	Fortress Chess	14
4.2	Horde Chess	14
4.3	Maharajah Chess	15
4.4	Amazon Chess	16
4.5	Úpravy ohodnocovacej funkcie pre jednotlivé varianty	16
5	Implementácia programu	19
5.1	Štruktúra aplikácie	19
5.2	Zaujímavé implementačné riešenia	20
5.2.1	Ohodnocovacia funkcia	20
5.2.2	Získanie legálnych ťahov hráča	20
5.2.3	Zistenie konca hry	21
5.3	Grafické užívateľské rozhranie	21
6	Záver	24
A	Obsah CD	26

Kapitola 1

Úvod

História šachu siaha až do dôb starovekých civilizácií. Jeho predchodcami si krátili čas ľudia po celom svete, vrátane Indie, Perzie, Európy, či Ruska. Moderná forma šachu bola sformovaná niekedy koncom 15. storočia v Taliansku a Španielsku. V priebehu nasledujúcich rokov boli ešte dolaďované rôzne drobné zmeny v pravidlách, kým sa hra nevyvinula do podoby, ktorú poznáme dodnes.

V priebehu vývoja šachu sa tvorili jeho rôzne varianty, ktoré mali za účel hru ozvláštniť a donútiť hráčov osvojiť si nové stratégie hrania. Tieto varianty šachu môžu nadobúdať rôzne podoby. Medzi inými môžeme spomenúť napríklad varianty s odlišnou hracou plochou, varianty s rôznym počiatočným rozmiestnením kameňov, alebo varianty s odlišnou sadou kameňov. Touto poslednou variantou sa budeme zaoberať v tejto práci.

Cieľom tejto práce je vytvoriť počítačový program, ktorý bude schopný osvojiť si rôzne stratégie hrania spojené s jednotlivými šachovými variantami. Tento program tiež poskytne prostriedky pre hru človeka proti človeku.

V prvej časti tejto práce sa pozrieme na hlavné vlastnosti šachu ako hry. Popíšeme si hernú plochu, jednotlivé figúrky, a vysvetlíme si pravidlá.

V druhej časti analyzujeme problematiku tvorby šachového programu. Rozoberieme jednotlivé štádia jeho návrhu a popíšeme si rôzne druhy mechanizmov používaných pri jeho tvorbe. Bližšie sa pozrieme na algoritmy používané pri vytváraní umelej inteligencie a ich aplikovanie na šach. Popíšeme si ohodnocovaciu funkciu pozícií pre klasický šach.

V tretej časti sa detailne zaoberáme štyrmi zvolenými variantmi šachu. Skúmame zmeny vyhodnocovania pozícií oproti klasickému šachu a navrhujeme zmeny do ohodnocovacej funkcie.

Vo štvrtej a poslednej časti práce sa budeme zaoberať návrhom samotnej aplikácie, popíšeme si rozdelenie aplikácie do tried, zvolené šachové štruktúry a implementáciu použitých mechanizmov. Popíšeme si grafické rozhranie programu a upresníme si jeho ovládanie.

Kapitola 2

Šach

2.1 História šachu

Obsah kapitoly je prebratý z [1]

Úplné začiatky šachu siahajú do 6. storočia k starej indickej hre známej ako *čaturanga*. Z tejto hry sú odvodené základné figúry šachovej hry:

- pechota - pešiaci
- jazda - jazdci
- slony - strelci
- vojenské vozy - veže

Čaturanga bola neskôr perzskými a arabskými obchodníkmi privezená do oblasti Stredného Východu, kde sa postupne vyvinula do hry známej ako *šatrandž*. Práve v Perzii je hra po prvýkrát označovaná pod pojmom šach, čo v perzštine znamená *kráľ*.

V 10. storočí sa hra cez maurských obchodníkov dostala do južnej Európy, kde si získala veľkú popularitu v Španielsku a Taliansku. Popularita šachu sa v nasledujúcich storočiach začala šíriť do celej Európy, šachoví majstri formulovali stratégie pre zahájenia a koncové hry, a v roku 1843 šachoví veľmajstri Tassilo von der Lasa a Paul Rudolf von Bilguer vydali prvý manuál šachovej teórie.

V roku 1924 bola založená Medzinárodná šachová federácia FIDE a začali sa organizovať prvé veľké šachové turnaje. Medzi najznámejších šachových veľmajstrov patria napríklad Gary Kasparov, Boris Spasskij alebo Bobby Fisher. Toto obdobie sa tiež označuje ako Vek veľmajstrov.

2.2 Pravidlá šachu

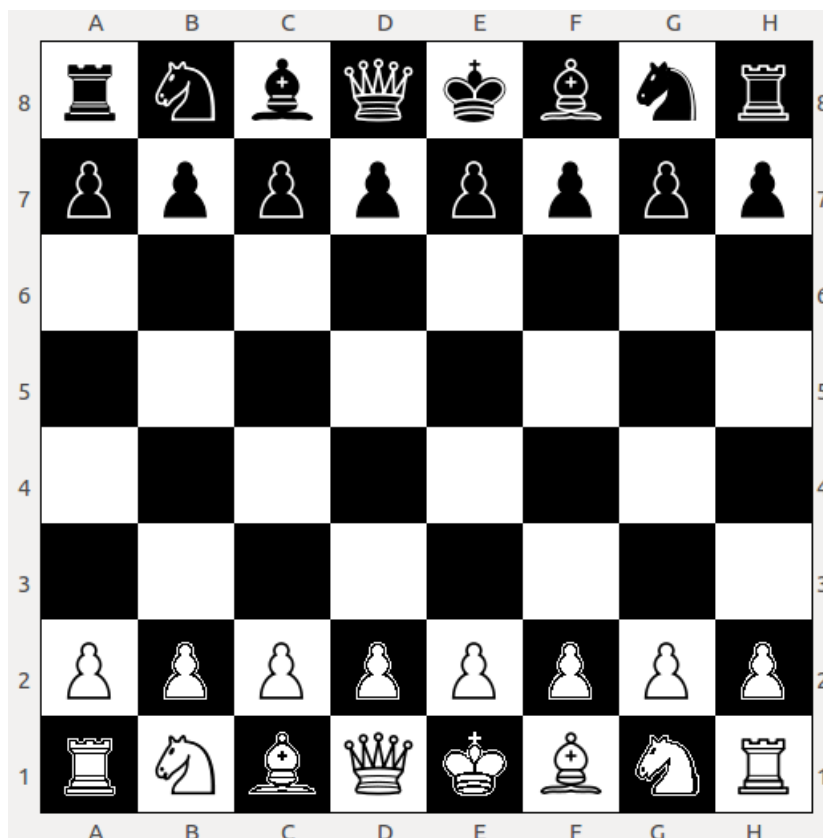
Šach je hra pre dvoch hráčov. Každý hráč hrá so sadou šestnástich kameňov, rozdelených do dvoch kategórií: osem pešiakov a osem figúr (dve veže, dvaja jazdci, dvaja strelci, dáma a kráľ). Jedna sada kameňov má bielu farbu, druhá čiernu. Podľa farby kameňov sú označovaní aj jednotliví hráči.

Hra sa odohráva na šachovnici pozostávajúcej z ôsmich radov a ôsmich stĺpcov štvorcov striedavo zafarbených na čierne a na bielo, čiže spolu 64 polí.

Hráči sa striedajú vo vykonávaní ťahov. Ťah pozostáva z presunu kameňa z jedného štvorca na iný (výnimkou je špeciálny ťah zvaný *rošáda*, pri ktorom sa hýbu dva kamene naraz). Hráč môže vo svojom ťahu zajať kameň protivníka, zajatie však nie je povinné.

Na začiatku hry sú kamene rozostavené nasledovne: druhý rad (z pohľadu daného hráča) je obsadený pešiakmi, v prvom rade sú zľava doprava umiestnené nasledovné figúry: veža, jazdec, strelec, dáma, kráľ, strelec, jazdec, veža. Za povšimnutie stojí, že dáma stojí vždy na poli svojej farby (biela dáma na bielom poli, čierna na čiernom).

Šachovnica a počiatočné rozostavenie figúr sú vyobrazené na obrázku 2-1.



Obrázek 2.1: Šachovnica s počiatočným rozostavením kameňov

V ďalšej časti tejto kapitoly si prejdeme jednotlivé kamene a popíšeme si ich možnosti pohybu:

- **Veža** je figúra, ktorá sa môže pohnúť o ľubovoľný počet polí vo vodorovnom alebo zvislom smere. Nedokáže preskakovať súperove kamene. Môže sa zúčastniť rošády.
- **Jazdec** je figúra, ktorá sa môže pohybovať tzv. do L, tzn. o dve polia vo vodorovnom/zvislom smere a o jedno pole v zvislom/vodorovnom smere. Dokáže preskakovať súperove kamene, jeho pohyb je obmedzený len tým, že nemôže doskočiť na políčko obsadené figúrou svojej farby.
- **Strelec** je figúra, ktorá sa môže pohnúť o ľubovoľný počet polí po diagonále. Nedokáže preskakovať súperove kamene.

- **Dáma** v sebe kombinuje pohybové možnosti veže a strelca, tzn. môže sa pohnúť o ľubovoľný počet polí vo vodorovnom alebo zvislom smere alebo po diagonále. Nemôže preskakovať súperove kamene.

- **Pešiak** sa za normálnych okolností môže pohnúť iba o jedno pole dopredu. V prípade, že stojí na druhom (počiatočnom) rade, môže postúpiť až o dve polia. Pešiak zajíma súperove kamene pohybom o jedno pole po diagonále dopredu.

Pešiaka sa týka jeden zo špeciálnych pohybov, tzv. *en passant*. Ak pešiak stojí v piatom rade (z pohľadu daného hráča) a súperov pešiak v príslušnom stĺpci postúpi z počiatočnej pozície o dve polia dopredu, pešiak môže súperovho pešiaka zajať skokom na pole, na ktorom by súperov pešiak stál, keby sa pohol iba o jedno pole. Tento ťah môže pešiak vykonať iba v ťahu bezprostredne po ťahu daného súperovho pešiaka.

V prípade, že sa pešiakovi podarí dostať na ôsmy rad (z pohľadu daného hráča), je povýšený na figúru (dáma, veža, jazdec, strelec) podľa vlastného výberu. Výber nie je obmedzený počtom figúr na šachovnici, tzn. hráč môže mať v jednej chvíli viac figúr jedného druhu než na začiatku hry.

- **Kráľ** sa môže pohnúť v ktoromkoľvek smere, tj. zvisle, vodorovne, alebo po diagonále, ale iba o jedno pole. Kráľ ako najdôležitejšia figúra hry sa nesmie pohnúť na pole ohrozené súperovým kameňom.

Druhý špeciálny pohyb v hre sa nazýva *rošáda* a vykonáva sa medzi kráľom a vežou jednej farby. Rozlišujeme malú a veľkú rošádu. Pri malej rošáde sa kráľ pohne o dve polia doprava a veža sa pohne o dve polia doľava. Pri veľkej rošáde sa kráľ pohne o dve polia doprava a veža sa pohne o tri polia doľava.

Možnosť rošády je podmienená splnením niekoľkých podmienok:

- Kráľ sa za celú hru nepohol z počiatočnej pozície.
- Veža, s ktorou kráľ rošádu vykonáva, sa za celú hru nepohla z počiatočnej pozície.
- Kráľ v čase vykonávania rošády nie je v šachu.
- Polia, cez ktoré kráľ počas vykonávania rošády prechádza, nie sú ohrozené protivníkovými kameňmi.
- Kráľ po vykonaní rošády neskončí na poli ohrozenom protivníkovými kameňmi.
- Všetky polia medzi kráľom a vežou, s ktorou rošádu vykonáva, sú voľné.

V priebehu šachovej partie môže dôjsť k nasledovným dôležitým udalostiam:

- **Šach** - V prípade, že je kráľ niektorého z hráčov ohrozený súperovým kameňom, hovoríme, že kráľ je v šachu. Vo svojom najbližšom ťahu musí tento hráč vykonať taký ťah, aby po jeho vykonaní kráľ v šachu nebol.
- **Mat** - Mat, alebo šach-mat nastáva v situácii, keď sa kráľ z niektorého z hráčov ocitne v šachu a neexistuje žiadny pohyb z jeho strany, ktorým by sa zo šachu dostal. Dosiahnutie tohto stavu je cieľom celej partie, a hráč, ktorý dá mat súperovmu kráľovi, partiu vyháva.
- **Pat** - Pat nastáva v prípade, že hráč na ťahu nemôže vykonať žiadny legálny ťah. Táto situácia nastáva spravidla vtedy, keď hráčov kráľ síce nie je v šachu, ale akýkoľvek jeho ťah by ho do ťahu dostal. V prípade patu sa partia končí remízou medzi hráčmi.

Kapitola 3

Tvorba šachového programu

Pri tvorbe šachového programu s implementáciou umelej inteligencie musíme vykonať niekoľko rozhodnutí, podľa ktorých budeme následne program implementovať. K týmto rozhodnutiam patrí:

- spôsob reprezentácie hernej plochy - akým spôsobom (datová štruktúra, poledots) budeme reprezentovať hernú plochu (šachovnicu). Pri tomto rozhodovaní kladieme dôraz na efektivitu prístupu k jednotlivým poliam a pamäťovú úspornosť reprezentácie.
- prehľadávanie stavového priestoru - algoritmus, pomocou ktorého budeme prehľadávať stavový priestor a vyvodzovať rozhodnutia ohľadne nasledujúcich ťahov. V spojitosti so šachom sa najčastejšie skloňuje algoritmus MiniMax a jeho rozšírenie Alfa-Beta.
- ohodnotenie jednotlivých stavov stavového priestoru - heuristika ohodnotenia je jadrom umelej inteligencie v šachovom programe. Existuje mnoho rôznych heuristík, z ktorých niektoré kladú dôraz na hrubú silu, zatiaľ čo iné uprednostňujú strategickjší prístup.

V tejto kapitole sa ďalej budeme venovať týmto trom bodom a vykonáme rozhodnutia, pomocou ktorých prikáročíme k implementácii programu.

3.1 Spôsob reprezentácie hernej plochy

Šachovnicu je možné reprezentovať niekoľkými spôsobmi, z ktorých najpoužívanéjšie si popíšeme v tejto časti:

- dvojrozmerné pole - výhodou tohto prístupu je prirodzenosť, osem krát osem políčok na šachovnici je reprezentovaných dvojrozmerným poľom s osem krát osem prvkami. Tento prístup je však prinajlepšom naivný. Jeho nevýhodou je veľká nepraktickosť pri programovaní i jeho neefektívnosť.
- jednorozmerné pole - polia sú uložené do 64-prvkového jednorozmerného poľa. Výhodou je oveľa efektívnejší prístup než u dvojrozmerného poľa, vďaka skutočnosti, že pre prístup k prvkom nám stačí jediný index. Nevýhodou zostáva fakt, že pri pohybe kameňov po šachovnici musíme overovať hranice hracej plochy, aby sa nám kameň "nepresunul" z jednej strany šachovnice na druhú.

- jednorozmerné pole s hraničnými prvkami - pole sa skladá zo 64 prvkov pre reprezentáciu polí na šachovnici, plus 80 prvkov pre dva štvorce hrubú hranicu na okrajoch plochy. Tieto prvky sú inicializované na určitú hodnotu, pomocou ktorej vieme, že posun na toto miesto je neplatný. Výhodou je očividne eliminácia nutnosti počítania, či index políčka nepretiekol na opačnú stranu hracej plochy. Nevýhodou je viac ako dvojnásobná veľkosť poľa.
- bitmapová reprezentácia - herná plocha je reprezentovaná 64-bitovým celým číslom, ktorého bity predstavujú jednotlivé políčka. Prístup k jednotlivým kameňom je potom implementovaný pomocou inštancií tohto čísla, kde jednotkové bity signalizujú výskyt danej figúrky. Tento prístup je veľmi efektívny predovšetkým na 64-bitových architektúrach, kde sú operácie so 64-bitovými číslami značne optimalizované, je však značne neintuitívny a jeho pochopenie môže trvať veľmi dlhú dobu.

Pre účely tejto práce som si pre reprezentáciu hernej plochy zvolil jednorozmerné pole štruktúr. Uvážil som, že tento prístup je kompromisom medzi efektívnosťou a zložitou implementáciou, a že pre účely tejto práce je dostatočný.

Jednotlivé herné polia sú reprezentované datovou štruktúrou, ktorá obsahuje položky ako kameň na políčku, farba políčka, pozícia políčka na šachovnici a pod. Konkrétnejší pohľad na samotnú implementáciu poskytuje kapitola 4.

3.2 Prehľadávanie stavového priestoru

Ako u všetkých hier pre dvoch hráčov pravidelne sa striedajúcich na ťahu s možnosťou jednoznačného ohodnotenia stavu hry po vykonaní každého ťahu, aj pri riešení umelej inteligencie pre šach je najvýhodnejšie použiť metódy hrania hier.

Vzhľadom k tomu, že v šachu je obrovské množstvo možných pozícií rozmiestnenia kameňov, nie je možné použiť jednoduché úplné prehľadanie stavov uložených do AND/OR grafu a vybrať ťah, ktorý bude naisto viesť k víťazstvu. Preto sa pri zložitejších hrách, ku ktorým patrí aj šach, používa vyhľadávanie najlepšieho možného ťahu na základe dopredu daného počtu ťahov oboch hráčov.

Nevyhnutnou súčasťou tohto prístupu je tzv. ohodnocovacia funkcia. Táto funkcia je diskutabilne najdôležitejšou súčasťou algoritmu pre hranie hier, pretože pomocou nej sa určuje hodnota vykonaných ťahov a vďaka tomu je možné vybrať najlepší nasledujúci ťah.

V tomto prípade sa javí ako najvýhodnejšie použiť algoritmus MiniMax a aplikovať naň rozšírenie Alfa-Beta.

Informácie o algoritme MiniMax a rozšírení Alfa-Beta sú prevzaté z [2] a informácie o heuristikách pre vylepšenie Alfa-Bety sú prevzaté z [6] a [7].

3.2.1 MiniMax

MiniMax je štandardnou metódou pre hranie hier. Pre nájdenie najlepšieho možného ťahu využíva hrubú silu, tzn. ohodnotí každú jednu možnú kombináciu ťahov oboch hráčov.

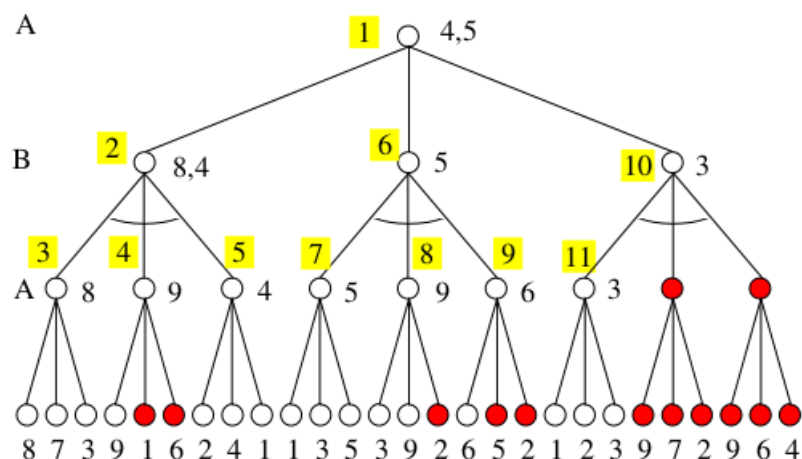
Princíp MiniMaxu spočíva v tom, že pri prehľadávaní stavového priestoru sa pre hráča na ťahu hľadá najlepší možný ťah, a pre jeho súpera ten najhorší. Z toho je odvodený aj názov MiniMax (hráč na ťahu je Max, jeho protivník je Min.)

Algoritmus MiniMax pracuje nasledovne:

1. Ak je aktuálny uzol stromu listom (dosiahla sa stanovená hĺbka zanorenia), ohodnot uzol ohodnocovacou funkciou a ohodnotenie vráť.

2. Ak je na ťahu hráč Max, zavolaj algoritmus MiniMax na každý z potomkov aktuálneho uzla a vráť najvyššiu získanú hodnotu. Ak je aktuálny uzlom koreňovým uzlom stromu, vráť cestu, ktorou si sa k najvyššej hodnote dostal.
3. Ak je na ťahu hráč Min, zavolaj algoritmus MiniMax na každý z potomkov aktuálneho uzla a vráť najnižšiu získanú hodnotu.

Príklad priebehu algoritmu MiniMax je vyobrazený na obrázku 3-1.



Obrázek 3.1: Priebeh algoritmu MiniMax

Na prvý pohľad vidíme, že algoritmus MiniMax síce po vyhodnotení všetkých možností vráti najlepší možný výsledok, jeho efektivita je však vo veľkej väčšine prípadov pomerne nízka. V obrázku sú červenou farbou vyznačené stavy, ktoré algoritmus vyšetroval zbytočne vzhľadom k tomu, že už ohodnocovanie ďalších pozícií lepši výsledok neprinesie. Čas potrebný na vyšetrenie všetkých pozícií tak s rastúcou hĺbkou zanorenia rastie exponenciálne. V tabuľke 3-1 vidíme počet ohodnocovaných uzlov podľa hĺbkou zanorenia pri priemernom počte potomkov na uzol 40.

Hĺbka	Počet možností
1	40
2	1,600
3	64,000
4	2,560,000
5	102,400,000

Tabulka 3.1: Počet ohodnocovaných uzlov

Práve z dôvodu nutnosti zníženia počtu prehľadávaných stavov a možnosti zvyšovať hĺbku výpočtu bolo zavedené "vylepšenie" algoritmu MiniMax s názvom Alfa-Beta.

3.2.2 Rozšírenie Alfa-Beta

Rozšírenie Alfa-Beta zavádza do problematiky metód hrania her pojem rez, konkrétne rez α a rez β . Rez α slúži na obmedzenie počtu prehľadávaných stavov hráča Max a rez β slúži

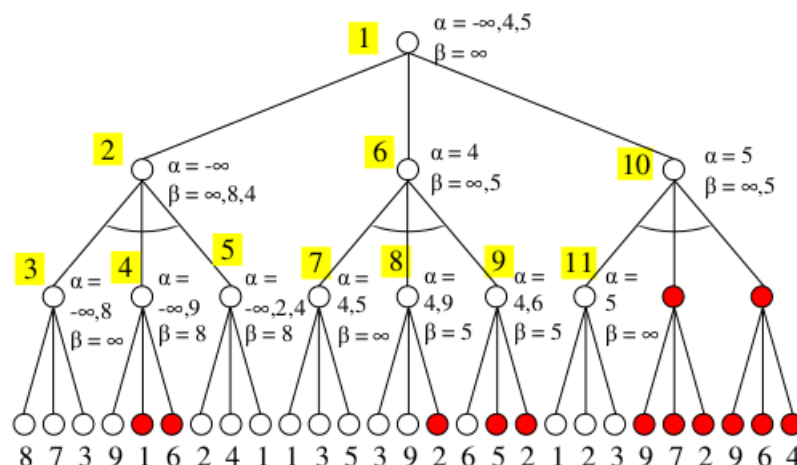
na obmedzenie počtu prehľadávaných stavov hráča Min.

Princíp tohto rozšírenia je ten, že algoritmus priebežne zaznamenáva najlepší získaný výsledok a v prípade, že niektorá vetva výpočtu v žiadnom prípade nevráti lepší výsledok, vetva sa oreže, tj. ďalej sa s ňou nepočíta.

Algoritmus MiniMax s rozšírením AlfaBeta funguje nasledovne:

1. Ak je aktuálny uzol koreňovým uzlom stromu, nastav hodnoty α na $-\infty$ a β na ∞ . V praxi to znamená, že si tieto hodnoty nastavíme tak, aby ich ohodnocovacia funkcia nedokázala prekročiť
2. Ak je aktuálny uzol listom stromu, ohodnoť uzol ohodnocovacou funkciou a ohodnotenie vráť.
3. Ak je na ťahu hráč Max, tak:
 - (a) Kým je $\alpha < \beta$, postupne volaj algoritmus so súčasnými hodnotami α a β na potomkov aktuálneho uzlu. Po každej iterácii nastav hodnotu α na maximum zo súčasných α a vrátenej hodnoty.
 - (b) Preruš cyklus a vráť aktuálnu hodnotu α . Ak je aktuálny uzol koreňovým uzlom stromu, vráť aj cestu k získanej hodnote α .
4. Ak je na ťahu hráč Min, tak:
 - (a) Kým je $\alpha < \beta$, postupne volaj algoritmus so súčasnými hodnotami α a β na potomkov aktuálneho uzlu. Po každej iterácii nastav hodnotu β na minimum zo súčasných β a vrátenej hodnoty.
 - (b) Preruš cyklus a vráť aktuálnu hodnotu β .

Príklad priebehu algoritmu MiniMax s použitím rozšírenia Alfa-Beta je vyobrazený na obrázku 2-2.



Obrázek 3.2: Priebeh algoritmu MiniMax

Z obrázku na prvý pohľad vidíme, že počet prehľadávaných stavov sa znížil zo 27 na 40, teda asi o štvrtinu (uzly vynechané z výpočtu sú na obrázku vyznačené červenou farbou).

Vzhľadom ku vlastnostiam Alfa-Bety môžeme vyvodiť záver, že správne usporiadanie prehľadávaných uzlov môže viesť k omnoho rozsiahlejšiemu orezávaníu zbytočne prehľadávaných ťahov a tým k veľkému zvýšeniu rýchlosti algoritmu.

Pre tento účel bolo vyvinutých niekoľko heuristik, ktoré nám usporiadávajú množinu uzlov vstupujúcu do algoritmu. Niekoľko z nich si popíšeme v nasledujúcej podkapitole.

3.2.3 Heuristiky pre zefektívnenie MiniMaxu a Alfa-Bety

- **Zožer, čo môžeš** - Táto jednoduchá heuristika je založená na tom, že ak daný ťah zajme súperov kameň, posunie sa bližšie k začiatku prehľadávaných ťahov. Môžeme priradiť bonus za rozdiel hodnôt zájmaného a zájmacieho kameňa.

Na prvý pohľad sa táto heuristika môže zdať nezmyselná, pretože zajatie súperovho kameňa nemusí vždy znamenať výhodu. Skutočnosť je ale taká, že hlbšie zanorenie drvivú väčšinu týchto nevýhodných ťahov rýchlo odhalí a vyberie tie najvýznamnejšie.

- **Historická heuristika** - Táto heuristika je založená na tom, že hodnoty jednotlivých pozícií sa medzi ťahmi súpera väčšinou líšia iba minimálne, preto je veľká šanca, že ak bol ťah dobre hodnotený pri jednej možnosti súperovho ťahu, bude dobre hodnotený aj pri inom.

Hlavnou otázkou tejto heuristiky je to, ako uchovávať ohodnotenia jednotlivých ťahov. Predstavíme si niekoľko možností:

- **Globálna tabuľka ťahov** Program uchováva tabuľku hodnôt jednotlivých ťahov. Hodnota ťahu sa zvýši v prípade, ak je ťah vyhodnotený ako dobrý, čo najčastejšie znamená, že vylepšil hodnotu α . Hodnota, o ktorú sa ohodnotenie ťahu zvýši môže byť upravené podľa hĺbky zanorenia (skúmané ťahy vo väčšej hĺbke majú logicky menší vplyv na počítaný najlepší ťah ako ťahy v nižšej hĺbke). Nevýhodou tejto metódy môže byť vysoká pamäťová náročnosť a zložitá réžia.
- **Najlepšie ťahy pre danú hĺbku** Program uchováva niekoľko najlepších ťahov pre každú hĺbku zanorenia. Tieto ťahy dostanú v nasledujúcej iterácii v rovnakej hĺbke prednosť. Veľkou výhodou je to, že touto metódou získame relevantnejšie výsledky, pretože uchováваме iba ťahy v hĺbke bezprostredne po spracúvanom ťahu. Nevýhodou je to, že si pamätáme iba pevný počet najhlbších ťahov.
- **Hlavná varianta** Program uchováva hlavnú variantu, to znamená cestu od listu až ku koreňu, ktorá mu vrátila najlepšie hodnotenie. V prípade, že sa v určitej hĺbke ohodnotenie zlepší, prepíšu sa len ťahy od danej hĺbky nižšie.
- **Dopočet do tichej pozície** Princípom tejto heuristiky je to, že listy grafu ohodnocujeme okrem klasickej ohodnocovacej funkcie aj upravenou Alfa-Beta metódou, ktorá berie do úvahy iba zájmanie kameňov a zvyšovanie pešiakov. Ako hodnotu listu vrátíme maximum z týchto dvoch hodnôt. Táto heuristika výrazným spôsobom napomáha ohodnocovaniu ťahov, ktoré vedú k výmene kameňov, a zvyšujú stabilitu prepočtu. Veľkou nevýhodou je to, že brzdí výpočet najlepšieho ťahu, pretože pre každý list musíme vykonať nový algoritmus MiniMax s Alfa-Betou.

3.3 Ohodnocovacia funkcia

Ako sme už spomínali, ohodnocovacia funkcia tvorí jadro výpočtu najlepšieho ťahu. Táto funkcia by mala vracať natoľko relevantné výsledky, aby už pri pomerne nízkej hĺbke vý-

počtu dokázal algoritmus MiniMax vyvodiť použiteľný výsledok.

Pri tvorení ohodnocovacej funkcie v aplikácii som sa inšpiroval ohodnocovacími funkciami v programoch GNU Chess [3] a Honzovy šachy [5].

Výpočet ohodnocovacej funkcie je rozdelený na dve hlavné časti - výpočet pozičnej zložky a výpočet materiálnej zložky. Materiálna zložka má spravidla vyššiu váhu ako pozičná zložka, vzhľadom k tomu, že strata figúry je pre hráča väčšinou horšia než zlé postavenie kameňov. Faktorom pri výpočte je tiež stav hry, teda či sa jedná o začiatok partie, stredovú hru, alebo zakončovanie. Tento faktor sa určuje spočítaním figúr zostávajúcich na šachovnici a ich typu.

Výsledkom ohodnocovacej funkcie je rozdiel medzi ohodnotením pozície z pohľadu jedného hráča a ohodnotením pozície z pohľadu druhého hráča.

3.3.1 Materiálna zložka

Hodnota jednotlivých figúr je zoradená podľa ich pohybovej schopnosti a celkovej užitočnosti v hre. V našej ohodnocovacej funkcii sme zvolili nasledovné hodnoty figúr:

Kameň	Hodnota
Pešiak	1000
Jazdec	3200
Strelec	3300
Veža	5500
Dáma	9500

Tabulka 3.2: Ohodnotenie kameňov v materiálnej zložke ohodnocovacej funkcie

Do materiálnej zložky je započítaný aj faktor, že dvaja strelci sú hodnotnejší ako súčet dvoch samostatných strelcov. Ak má hráč oboch strelcov, je tento fakt hodnotený bonusom.

3.3.2 Pozičná zložka

Pre každý z kameňov sú započítané nasledovné hodnoty:

- **Kráľ:**

- Ak je začiatok partie, hodnota kráľa je určená tabuľkou 3-3

Z tabuľky vidíme, že kráľ je najlepšie hodnotený na zadných radoch, na poliach, na ktoré sa dá dostať rošádou. Naopak veľmi záporne hodnotený je pohyb kráľa dopredu.

-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000
-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000
-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000
-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000
-1000	-1000	-1000	-1000	-1000	-1000	-1000	-1000
-500	-500	-500	-500	-500	-500	-500	-500
0	-200	-300	0	0	-300	-200	0
100	450	400	0	0	0	500	100

Tabulka 3.3: Tabuľka pozičnej zložky kráľa

- Ak súper zakončuje, kladne sa hodnotí blízkosť kráľa ku stredu šachovnice a vzdialenosť kráľov od seba.
- Záporne sa hodnotí, ak je hráčovi znemožnená rošáda.

- **Dáma:**

- Ak hráč zakončuje, kladne sa hodnotí blokovanie kráľa na kraji šachovnice.
- Dáma iné pozičné ohodnotenie nemá, jej vysoká materiálna hodnota jej bráni zbytočne vojsť na ohrozené pole.

- **Veža:**

- U veže je kladne hodnotené, ak sa nachádza v stĺpci bez pešiakov.

- **Strelec:**

- Hodnota strelca je určená tabuľkou 3-4
Z tabuľky vidíme, že strelce je najlepšie hodnotený na pozícii pred kráľom v strede radu, kde môže kráľa kryť a zároveň môže ohroziť čo najviac polí. Naopak nízko hodnotené sú pozície v rohoch šachovnice, kde je jeho rozhľad obmedzený.

-30	-30	-20	-20	-20	-20	-30	-30
-10	20	40	60	60	40	20	-10
20	30	50	70	70	50	30	20
50	200	50	50	50	50	200	50
50	20	200	20	20	200	20	50
-20	-20	-20	100	100	-20	-20	-20
20	200	-20	100	100	-20	200	20
0	-20	0	-200	-200	0	-20	0

Tabuľka 3.4: Tabuľka pozičnej zložky strelca

- **Jazdec:**

- Hodnota jazdca je určená tabuľkou 3-5
Strelec je logicky najlepšie hodnotený v strede šachovnice, kde jeho nezvyčajné pohybové vlastnosti dokážu naplno vyniknúť. Celkom dobre je hodnotené aj umiestnenie pred kráľom, kde mu môže slúžiť ako štít. Naopak veľmi negatívne je ohodnotené postavenie v rohoch, kde môže využiť iba dva z ôsmich možných pohybov.

- **Pešiak:**

- Hodnota pešiaka je určená tabuľkou 3-6
Pešiáci sú kladne hodnotení vo vyšších radoch na ceste k povýšeniu. Tento faktor je posilnený bonusmi za to, keď má pešiak voľnú cestu k ôsmemu radu a je krytý. Posledný rad je hodnotený nulami preto, že pešiak v poslednom rade už nie je pešiakom, ale mení sa na figúru. Prvý rad je hodnotený nulami preto, že pešiak sa tam nikdy nedostane.
Významné je aj vysoké hodnotenie postavenia pešiakov v stredných stĺpcoch uprostred šachovnice, kde môžu slúžiť ako hradba pred kráľom.

-150	0	0	50	50	0	0	-150
0	0	150	250	250	150	0	0
0	120	150	250	250	150	120	0
0	120	150	250	250	150	120	0
-50	0	80	200	200	80	0	-50
-100	-50	100	70	70	100	-50	-100
-200	-150	50	0	0	50	-150	-200
-300	-100	-150	-50	-50	-150	-100	-300

Tabulka 3.5: Tabuľka pozičnej zložky jazdca

0	0	0	0	0	0	0	0
100	100	100	100	100	100	100	100
200	200	200	200	200	200	200	200
-150	-100	150	250	250	0	-150	100
-200	-150	200	250	250	0	-250	-200
20	10	-150	50	50	-300	10	20
10	20	0	0	0	0	20	10
0	0	0	0	0	0	0	0

Tabulka 3.6: Tabuľka pozičnej zložky pešiaka

- Kladne sa hodnotí ak je pešiak krytý iným pešiakom, a ak sa nachádza v stĺpci bez pešiakov súpera.
- Počas strednej hry a koncovky sa kladne hodnotí, ak pešiak postupuje dopredu a snaží sa nechať povýšiť. Tento bonus je ďalej zvýraznený, ak je pešiak krytý a vo voľnom stĺpci.
- Záporne sa hodnotí, ak v susedných stĺpcoch k pešakovmu stĺpcu sa nenachádza hráčov pešiak.

Kapitola 4

Zvolené šachové varianty

Všetky šachové variácie spomenuté v tejto práci sú prevzaté z [8].

Pre účely tejto práce som si zvolil štyri varianty šachu s rozdielnou sadou figúr: Fortress Chess, Horde Chess, Maharajah Chess a Amazon Chess. V štyroch nasledujúcich podkapitolách si tieto varianty popíšeme. Ďalej v kapitole si prejdeme zmeny, ktoré bude nutné urobiť vo vyhodnocovacej funkcii klasického šachu pre efektívnu hru týchto variánt.

4.1 Fortress Chess

Fortress Chess sa od klasického šachu líši počtom pešiakov a počiatočným rozostavením figúr:

- Obaja králi sú umiestnení do pravého dolného rohu (z pohľadu príslušného hráča).
- Ostatné figúry (dáma, obidve veže, obaja strelci a obaja jazdci) sú náhodne rozmiestnené v prvom rade (z pohľadu príslušného hráča).
- Jeden strelec musí stáť na bielom poli a druhý na čiernom.
- Pešiaci zostávajú na svojej pôvodnej pozícii - druhý (siedmy) rad.
- Počiatočné pozície oboch hráčov sú symetrické podľa stredu. To znamená, že ak biela dáma stojí na poli C1, čierna dáma musí stáť na poli F8.
- Pred oboma kráľmi sú umiestnení traja pešiaci navyše.

4.2 Horde Chess

Táto šachová varianta sa od ostatných variánt líši na prvý pohľad. Biely hrá s normálnou sadou šachových kameňov, čierny má len hordu 32 pešiakov rozmiestnených presne ako na obrázku 4-2.

Pravidlá sú rovnaké ako u klasického šachu s nasledujúcimi výnimkami:

- Biely vyhráva, ak sa mu podarí zajať všetky čierne kamene. Čierny vyhráva, ak sa mu podarí dať šach-mat bielemu kráľovi.
- Ak žiadny čierny kameň nemôže vykonať platný pohyb, hra končí remízou.



Obrázek 4.1: Počiatočné rozostavenie Fortress Chess

- Čiernych pešiakov je možné povýšiť na čiernu dámu, vežu, jazdca alebo strelca, ak dosiahnu posledný rad (z pohľadu čierneho hráča).
- Čierni pešiáci sa nemôžu pohnúť o dve polia z žiadnej inej pozície než z druhého radu (z pohľadu čierneho hráča).

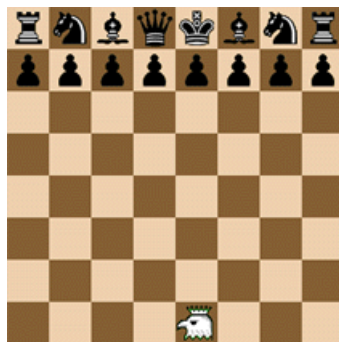


Obrázek 4.2: Počiatočné rozostavenie Horde Chess

4.3 Maharajah Chess

Táto neobvyklá variácia šachu, nazývaná tiež *Maharajah and the Sepoys* je takmer 300 rokov známa v Indii a na niekoľkých Malajských ostrovoch. Hra obsahuje novú figúru zvanú Maharadža (Maharajah), ktorá v sebe spája vlastnosti kráľa, dámy a jazdca. V praxi to znamená, že maharadža sa pohybuje ako kráľovná aj ako jazdec, nemôže vojsť na pole ohrozované kameňom nepriateľa, a ak je ohrozovaný kameňom nepriateľa, musí sa buď stiahnuť alebo ohrozujúcy kameň zajať. Počiatočné rozmiestnenie kameňov je na obrázku 4-3.

Ďalšie pravidlá (s jednou výnimkou - čierni pešiáci sa nemôžu povýšiť na žiadnu inú figúru a zostávajú v poslednom rade do konca hry) sú identické s pravidlami klasického šachu. To znamená, že hra končí výhrou bieleho/čierneho hráča, ak kráľ/Maharadža dostanú šach-mat, hra skončí remízou, ak hráč na ťahu nedokáže vykonať žiadny platný pohyb, atď.



Obrázek 4.3: Počiatočné rozostavenie Maharajah Chess

4.4 Amazon Chess

Amazon Chess je jednoduchou variáciou šachu, v ktorej je dáma nahradená novou figúrou zvanou "Amazonka". Je vyobrazená ako jazdec s korunkou dámy, čo aj naznačuje jej pohybové možnosti - Amazonka kombinuje pohyby dámy a jazdca, podobne ako Maharadža v Maharajah Chess. Všetky ostatné pravidlá sú rovnaké ako v klasickom šachu.



Obrázek 4.4: Počiatočné rozostavenie Amazon Chess

4.5 Úpravy ohodnocovacej funkcie pre jednotlivé varianty

Vzhľadom k tomu, že pri hraní variácií šachu je nutné brať do úvahy nové fakty, ako sú zmena kameňov alebo odlišné počiatočné rozmiestnenie, bolo nutné umelú inteligenciu programu adaptovať. Toho sme dosiahli úpravou ohodnocovacej funkcie. V ďalšej časti kapitoly si prejdeme jednotlivé šachové varianty a aké úpravy sme pre ne vykonali v ohodnocovacej funkcii.

- **Fortress Chess**

- Znížili sme hodnotu pešiaka na 800, kvôli ich zvýšenému počtu.
- Zrušili sme penalizáciu za zdvojeného pešiaka v stĺpcoch 6-8.
- Zrušili sme penalizáciu za znemožnenie rošády, pretože rošáda ako taká nie je v tejto variácii vôbec možná.

- Upravili sme hodnotu pešiakov pomocou novej tabuľky, v ktorej je odrazená zvýšená snaha troch nových pešiakov dostať sa na vzdialenú stranu šachovnice a nechať sa povýšiť na figúru.
- Novú tabuľku sme vytvorili aj pre kráľa.
V tejto tabuľke je lepšie ohodnotené uschovanie sa kráľa za pevnosť z novo postavených pešiakov.

• Horde Chess

- Hodnotu čierneho pešiaka sme upravili tak, aby bol súčet materiálnych zložiek oboch hráčov rovnaký.
- Zrušili sme penalizáciu za zdvojeného pešiaka v stĺpci.
- Čiernemu hráčovi sme zrušili penalizáciu za znemožnenie rošády.
- Vytvorili sme novú tabuľku hodnôt pre čiernych pešiakov. Táto tabuľka je zobrazená ako Tabuľka 4-1.

V tejto tabuľke je zahrnutá snaha čiernych pešiakov rovnomerne sa posúvať dopredu, a ak je niektorý stĺpec uvoľnený, dostať sa do posledného radu a nechať sa povýšiť na figúru. Po povýšení na figúru sa táto figúra správa rovnako ako jej náprotivok v klasickom šachu.

0	0	0	0	0	0	0	0
100	100	100	100	100	100	100	100
200	200	200	200	200	-200	200	200
-150	-100	150	250	250	0	-150	100
-200	-150	200	250	250	0	-250	-200
20	10	-50	50	50	-300	10	20
10	20	0	0	0	0	20	10
0	0	0	0	0	0	0	0

Tabuľka 4.1: Tabuľka pozičnej zložky čiernych pešiakov vo variante Horde Chess

• Maharajah Chess V tejto variante sme kvôli diametrálne odlišnej hre ohodnocovali funkciu z veľkej časti úplne prekopali.

- Maharadžovi sme prideliť hodnotu materiálnej zložky rovnú súčtu hodnôt materiálnej zložky kameňov čierneho hráča.
- Pri hre za maharadžu sa snažíme maximalizovať počet polí, na ktoré sa bude maharadža môcť po vykonaní presunúť.
- Takisto za každý kameň, ktorý maharadža ohrozí, si k hodnote ťahu pripočíta hodnotu tohto kameňa. Tým sa čierny hráč povzbudí, aby si dôležité figúry kryl a slepo sa nevrhal ohroziť maharadžu.
- Pri hre za čierneho hráča sme upravili heuristiku kráľa tak, že ak ešte hráč vlastní veľa figúr, bude jeho pozícia ohodnotená podľa upravenej tabuľky 3-10
Podľa tejto tabuľky je kráľ najlepšie hodnotený v strede. Je to z toho dôvodu, že v strede má najviac príležitostí uniknúť šachu zo strany maharadžu.

- Hodnota pešiakov je tiež upravená novou tabuľkou, ktorá je zobrazená ako Tabuľka 4-2

Táto tabuľka odráža fakt, že sa pešiaci po dosiahnutí posledného radu nepovyšujú, ale zostanú tam stáť do konca hry. Preto majú polia za polovicou šachovnice vysoko záporné hodnotenie.

- Odstránili sme tiež bonus pre pešiaka za uvoľnený stĺpec, pretože v tejto variante je jeho postup dopredu nežiadúci.
- Zvyšné čierne figúry sa snažia čo najviac obmedziť maharadžove možnosti pohybu, tzn. získavajú bonus za každé pole z maharadžových možných cieľov pohybu, ktoré majú pod svojou kontrolou.

-4000	-4000	-4000	-4000	-4000	-4000	-4000	-4000
-2500	-2500	-2500	-2500	-2500	-2500	-2500	-2500
-800	-700	-200	-200	-200	-200	-1000	-800
-500	-450	0	0	0	-100	-650	-500
-200	-150	200	250	250	0	-250	-200
20	10	-50	50	50	-300	10	20
10	20	0	0	0	0	20	10
0	0	0	0	0	0	0	0

Tabuľka 4.2: Tabuľka pozičnej zložky čiernych pešiakov vo variante Maharajah Chess

-400	-300	-200	-100	-100	-200	-300	-400
-300	-200	0	50	50	0	-200	-300
-200	0	50	100	100	20	0	-200
-100	50	100	200	250	0	-250	-200
-200	0	50	100	100	20	0	-200
-100	50	100	200	250	0	-250	-200
-300	-200	0	50	50	0	-200	-300
-400	-300	-200	-100	-100	-200	-300	-400

Tabuľka 4.3: Tabuľka pozičnej zložky čierneho kráľa vo variante Maharajah Chess

• Amazon Chess

- Jedinými úpravami pri tejto šachovej variácii bolo zvýšenie materiálnej hodnoty Amazonky na 12,000, zvýšenie bonusu pri tlaku na súperovho kráľa v koncovke a udelenie penalizácie za jej skoré rozvinutie.

Kapitola 5

Implementácia programu

Celá programová časť práce bola naimplementovaná pod názvom Chess2012 v jazyku C++ s využitím frameworku Qt4, prevzatého z [4]. Tento framework je na programovanie grafických aplikácií vhodný, pretože poskytuje prostriedky pre ich tvorbu a vývoj.

V prvej časti tejto kapitoly si popíšeme obsah jednotlivých zdrojových súborov, v druhej sa pozrieme na konkrétne implementačné riešenia zaujímavých častí v kóde.

5.1 Štruktúra aplikácie

Aplikácia pozostáva z deviatich zdrojových súborov, deviatich hlavičkových súborov, a štyroch formulárov vytvorených v aplikácii QtDesigner.

Zdrojové súbory sú tieto:

- **ai.cpp** - Súbor, v ktorom sa nachádza celá implementácia umelej inteligencie programu. V súbore sú definované dve triedy: **Ai**, ktorá implementuje algoritmus Minimax s rozšírením AlfaBeta, a **Evaluation**, ktorá obsahuje statické metódy implementujúce ohodnocovacie funkcie pre klasický šach i jednotlivé variácie.
- **board.cpp** - Súbor obsahujúci triedu **Board**, ktorá reprezentuje šachovnicu ako celok. Táto trieda obsahuje metódy pre vypočítanie možných ťahov z danej pozície, a spravuje polia šachovnice v jednorozmernom 64-prvkovom vektore obsahujúcom inšancie triedy **Square**.
- **game.cpp** - Súbor s triedou **Game**, ktorá obsahuje všetky dôležité funkcie pre riadenie samotnej hry. Obsahuje implementáciu metód pre predávanie ťahov medzi hráčmi, určenie platných ťahov hráča, overenie, či sa kráľ nenachádza v šachu, prípadne nastala remíza, a i.
- **main.cpp** - Súbor s hlavnou aplikačnou slučkou aplikácie.
- **mainwindow.cpp** - Súbor s triedou **MainWindow** predstavujúcou hlavné okno aplikácie. Obsahuje predovšetkým metódy s úlohou predávať informácie z riadiacich tried do dialógov a naopak.
- **newgamedialog.cpp** - Súbor spracúvajúci logiku formulára **NewGameDialog**.
- **player.cpp** - Súbor s triedou **Player**, ktorá reprezentuje každého z hráčov. Okrem tejto triedy obsahuje aj triedu **Figure**, ktorá predstavuje jeden hrací kameň.

- **promotedialog.cpp** - Súbor spracúvajúci logiku formulára `PromoteDialog`.
- **square.cpp** - Súbor obsahujúci triedu `Square` predstavujúcu jedno pole na klávesnici. Obsahuje napr. metódy pre premaľovanie poľa na určitý kameň či pridelenie figúry poľu. Spracúva tiež logiku kliknutia myši na pole, tzn. vyznačenie možných ťahov, výber poľa, na ktoré bude hráč ťahať, a zahájenie samotného presunu figúry.

5.2 Zaujímavé implementačné riešenia

5.2.1 Ohodnocovacia funkcia

Úlohou ohodnocovacej funkcie je zo vstupného parametra aktuálnej hry `Game *g` získať číselné ohodnotenie aktuálne overovaného ťahu.

Samotná inštancia triedy `Game` obsahuje všetky informácie, ktoré k tomuto výpočtu potrebujeme: šachovnicu, z ktorej získame aktuálne pozície kameňov, aktuálneho hráča, pre ktorého budeme hodnotu pozície počítať, a umožní nám i prístup k triednym metódam, akými je získanie legálnych ťahov alebo overenie, či sa hráč nenachádza v šachu.

Na úvod si postupným prejdением všetkých políčok šachovnice zistíme počet kameňov každého typu v hre. Z tohto počtu si následne odvodíme typ pozície (začiatok partie, stredová hra, koncovka).

Pre samotné počítanie ohodnotenia pozície si najprv inicializujeme pomocnú premennú `value` na hodnotu 0.

Po tejto inicializácii začneme po druhýkrát prechádzať všetky polia šachovnice a podľa toho, aký kameň sa na aktuálnom poli nachádza, upravíme hodnotu premennej `value` o číslo určené buď tabuľkami pre pozičnú zložku ohodnotenia alebo o dopredu stanovenú konštantu. V prípade, že je na poli kameň aktuálneho hráča, túto hodnotu pričítavame, v opačnom prípade ju odčítame.

Po prejení všetkých polí šachovnice máme premennú `value` nastavenú na pozičnú zložku ohodnotenia. Nakoniec k nej pripočítame materiálnu zložku, ktorú získame súčtom hodnôt jednotlivých kameňov vynásobených ich počtom.

Po pričítaní materiálnej zložky je premenná `value` nastavená na celkové ohodnotenie pozície a môžeme ju príkazom `return` odoslať do volajúcej funkcie `AlfaBeta`.

5.2.2 Získanie legálnych ťahov hráča

Základom pre vypočítanie legálnych ťahov hráča sú rutiny v triede `Board`:

- **createVertMoves** - získa všetky voľné políčka vo vertikálnom a horizontálnom smere od danej pozície. Rutina obsahuje zarážku pre jeden ťah do každého smeru v prípade, že na danej pozícii stojí kráľ.
- **createDiagMoves** - získa všetky voľné políčka po diagonálach od danej pozície. Rutina obsahuje zarážku pre jeden ťah do každého smeru v prípade, že na danej pozícii stojí kráľ.
- **knightMoves** - získa všetky voľné políčka do L od danej pozície
- **pawnMoves** - získa všetky políčka na ťah pre pešiaka, vrátane postupu o dve políčka z počiatočnej pozície a zajímania *en passant*.
- **kingCastlingMoves** - získa políčka, na ktoré je možné spraviť rošádu.

Samotné získanie všetkých legálnych ťahov hráča obsluhuje metóda `setLegalMoves` triedy `Game`. Táto trieda postupne prejde všetky polia šachovnice, a ak sa na aktuálnom poli nachádza kameň aktuálneho hráča, zavolá pre konkrétny kameň špecifické rutiny.

Ak je napríklad na políčku strelec, metóda zavolá rutinu `createDiagMoves`. Ak je na ňom dáma, zavolá rutiny `createVertMoves` aj `createDiagMoves`. Pre kráľa zavolá rutiny `createVertMoves`, `createDiagMoves`, aj `kingCastlingMoves`.

Týmto spôsobom sme získali všetky možné ťahy z danej pozície. Legálne ťahy z nich získame tak, že postupne každý ťah vykonáme, zavoláme metódu `checkForCheck` a ťah vrátíme. Ak sa po vykonaní ťahu kráľ hráča ocitne v šachu, ťah nie je legálny.

Metóda `checkForCheck` funguje tak, že získa ťahy protivníka, overí, či sa v cieľovom poli niektorého z nich nachádza pole s kráľom aktuálneho hráča. a túto hodnotu vráti.

5.2.3 Zistenie konca hry

Na začiatku každého ťahu si pre hráča na ťahu získame vektor legálnych ťahov metódou `setVectorMoves`. V prípade, že je tento vektor prázdny, znamená to, že hráč nemôže vykonať žiadny pohyb.

Ak sa tak stane, metódou `checkForCheck` zistíme, či sa hráčov kráľ práve nachádza v šachu. Ak áno, hra sa končí víťazstvom pre súpera. Ak nie, hra končí patom.

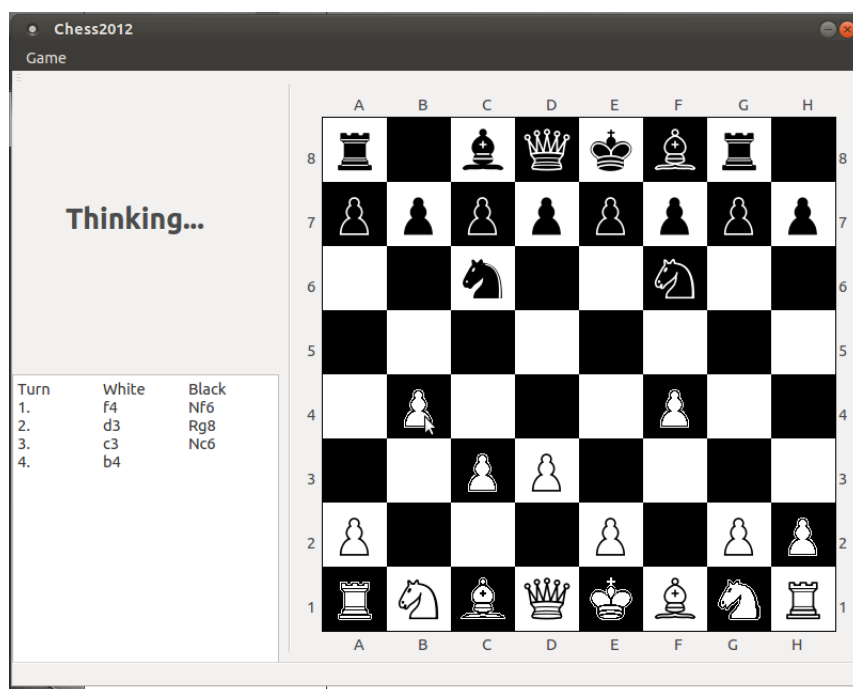
5.3 Grafické užívateľské rozhranie

Hlavné okno, v ktorom sa užívateľské rozhranie nachádza, je reprezentované triedou `MainWindow`. Nachádza sa v ňom šachovnica, ktorá zaberá väčšinu priestoru okna a je orientovaná doprava, ľavú dolnú časť zaberá textové pole, do ktorého sa postupne vypisujú odohrané ťahy v šachovej algebraickej notácii, a ľavý horný roh obsadzuje nápis *Thinking...*, ktorý sa zviditeľní vždy, keď je na ťahu počítačom ovládaný hráč a po odohraní jeho ťahu opäť zmizne.

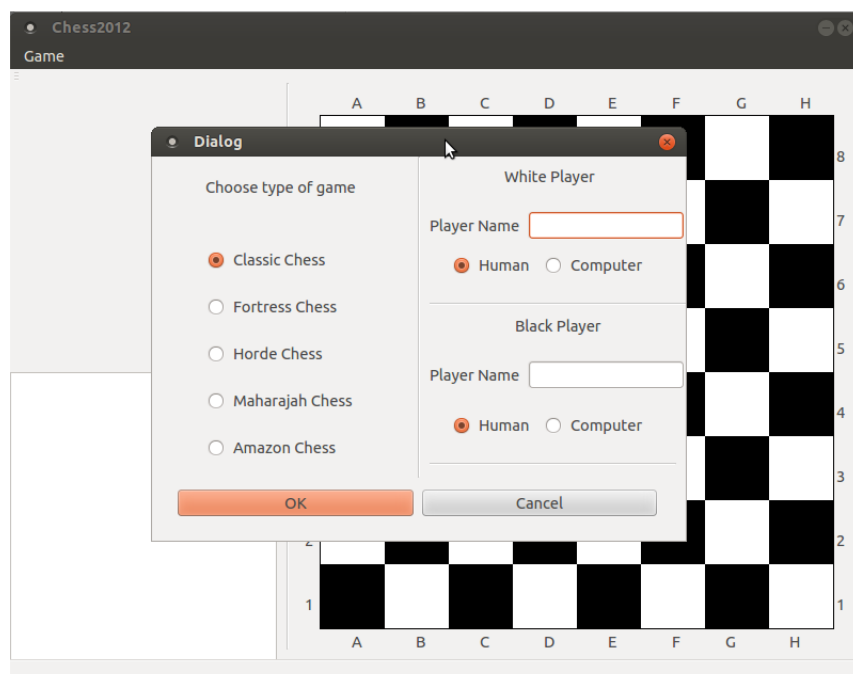
Druhým významným oknom užívateľského rozhrania je dialóg pre vytvorenie novej hry. Tento dialóg obsahuje v pravej časti polia pre zadanie mien hráčov a výber ich typu (ľudský hráč, alebo počítačom ovládaný hráč). V ľavej časti dialógu sa nachádzajú prepínacie tlačidlá pre výber šachovej variácie, ktorú chceme hrať.

Tretím a posledným oknom užívateľského rozhrania je dialóg pre povýšenie pešiaka na figúru. Tento dialóg sa zobrazí hneď potom, ako pešiak dorazí na posledný rad. Dialóg obsahuje štyri prepínacie tlačidlá s ikonkami jednotlivých figúr. Pešiak sa po uzavretí okna zmení na tú figúru, ktorej prepínacie tlačidlo bolo aktívne.

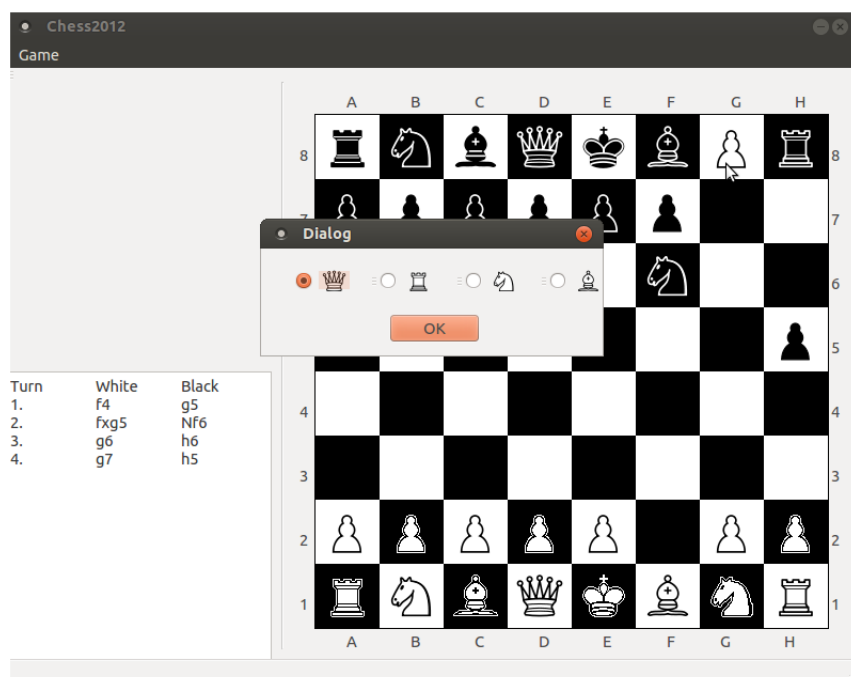
Jednotlivé okná užívateľského rozhrania môžeme vidieť na obrázkoch 5-1 až 5-3.



Obrázek 5.1: Hlavné okno aplikácie v akcii



Obrázek 5.2: Dialóg na vytvorenie novej hry v akcii



Obrázek 5.3: Dialóg na povýšenie pešiaka v akcii

Kapitola 6

Záver

Úlohou práce bolo navrhnuť a implementovať počítačový program, ktorý dokáže hrať šach a jeho rôzne varianty s rôznou sadou figúr. Výsledkom je aplikácia, ktorá spĺňa dané zadanie a predstavuje pre ľudského hráča začiatočníka dôstojného súpera. Aplikácia však pri súčasne danej hĺbke zanorenia nie je schopná myslieť na viac ako tri polťahy dopredu, preto sa profesionálnemu hráčovi nevyrovná.

Aplikácia poskytuje aj možnosť hrať hru človeka proti človeku na jednom počítači, pri ktorej si hráči striedavov vyberajú svoje ťahy.

Navrhovanými zlepšeniami do budúcnosti sú zefektívnenie mechanizmu výberu ťahov, použitie heuristik na zlepšenie efektivity metódy Alfa-Beta a samozrejme ďalšia analýza a vylepšenie ohodnocovacích funkcií.

Literatura

- [1] Gene Rodriguez, I.: A Brief Chess History.
<http://www.life123.com/hobbies/games/chess/history-of-chess.shtml>, 2012.
- [2] doc. Ing. František Zbořil CSc., I. F. Z. P.: *Základy umělé inteligence IZU, Studijní opora*. Fakulta informačních technologií, Vysoké učení technické, Brno, 2006.
- [3] Mathematics, I.; Academy, S.: GnuChess.txt.
<http://alumni.imsa.edu/~stendahl/comp/txt/gnuchess.txt>, 1987 [cit. 2012-05-16].
- [4] Nokia: Download Qt, the cross-platform application framework.
<http://qt.nokia.com/downloads>.
- [5] Němec, J.: Honzovy šachy.
<http://www.euroekonom.cz/programy-clanky.php?type=sachy>.
- [6] Němec, J.: Šachové myšlení (4) - vylepšení alfabety. 2006-07-17.
- [7] Němec, J.: Šachové myšlení (6) - Prořezávání.
http://www.linuxsoft.cz/article.php?id_article=1335, 2006-10-31.
- [8] Rachůnek, F.: BrainKing - Game rules. <http://brainking.com/en/GameRules>.

Příloha A

Obsah CD

Zdrojové kódy aplikace a dokumentácia v elektronickej forme.